A Scalable Model for Multi-period Virtual Network Mapping for Resilient Multi-Site Data Centers

Ting Wang and Brigitte Jaumard
Computer Science and Software Engineering
Concordia University
Montreal (QC) H3G 1M8 Canada

Chris Develder
INTEC – IBCN
Ghent University – iMinds
Ghent, Belgium

***Abstract*—In the currently dominant cloud computing paradigm, applications are being served in data centers (DCs) which are connected to high capacity optical networks. For cost efficiency reasons, in both DC and optical network domains, virtualization of the physical hardware is exploited. In a DC, it means that multiple so-called virtual machines (VMs) are being hosted on the same physical server. Similarly, the network is partitioned into separate virtual networks, thus providing isolation between distinct virtual network operators (VNOs). Thus, the problem of virtual network mapping arises: how to decide which physical resources to allocate for a particular virtual network? In this paper, we study that problem in the context of cloud computing with multiple DC sites. This introduces additional flexibility, due to the anycast routing principle: we have the freedom to decide at what particular DC location to serve a particular application. We can exploit this choice to minimize the required resources when solving the virtual network mapping problem. This paper builds on our earlier work and solves the resilient virtual network mapping problem that optimally decides on the mapping of both network and data center resources, considering time-varying traffic conditions and protecting against possible failures of both network and DC resources. Previously, we developed a model to solve the multi-period traffic case one step at a time: given the virtual network mapping in period t, we determine the (possibly changed) mapping for $t + 1$. Compared to that previous work, we now *(i)* define a truly multi-period model path formulation exploiting column generation, and *(ii)* demonstrate its scalability on a nation-wide network with traffic that varies across multiple periods.**

I. Introduction

Increasingly, businesses are relying on cloud computing: applications and content are being served in data centers (DCs). Given the high bandwidth capacity and low latency of the underlying optical networking technology, users typically do not care very much about the exact location of these data centers, and service providers can exploit anycast routing: to serve a new request, they basically have the freedom to pick any of the available data centers. As a result, this anycast principle can be exploited for resiliency purposes [1]: if either the server infrastructure (in the DC), or the (optical) network is affected by a failure, backup could be provided at a different DC location compared to failure-free operation. Earlier work has considered the potential benefits of exploiting anycast on the required network capacity under static traffic (e.g., [2]).

This paper considers time-varying traffic: we investigate the potential savings of reconfiguring traffic routing (working and/or backup routes) from one time period to the next, assuming a time-slotted approach where the traffic patterns change from one period to the next but some traffic does survive multiple periods. This topic has been investigated in the past, but not thoroughly. For instance, He and Poo [3] propose a sub-reconfiguration technique in order to rearrange the paths for WDM (Wavelength Division Multiplexing) networks, using pre-computed alternate backup paths. They report a 10% bandwidth saving with simulation experiments using OPNET. Other studies look at differentiated protection schemes, e.g., [4] or [5], with either pre-emption or multiple protection paths, but without backup reconfiguration. But these works considered "classical" traffic, i.e., not the cloud computing scenario with anycast routing. The current paper builds on our earlier work [6], which to our knowledge was the first to consider resilient multi-period anycast traffic routing. Yet, there we assumed an iterative approach: we formulated the optimization problem to find the best routing going from one period to the next. Our current work presents a model for joint optimization over all time periods together.

The paper is organized as follows. In Section II, we give an overview of resilience in cloud computing settings, where the physical infrastructure is usually shared by multiple virtual network operators (VNOs). We highlight the scenarios we consider for changing (or not) configurations that last multiple periods in time-varying traffic settings. In Section III, we present our new model for finding the routes (for each of the multiple time periods) that minimize the bandwidth requirements to serve time-varying cloud traffic. We

apply it to a case study to quantitatively compare the various scenarios of changing primary and/or backup routes in Section IV. We summarize our conclusions in Section V.

II. Problem Statement

A. Virtualization and Resilience in Cloud Computing

The recent evolution towards grid and cloud computing illustrates the crucial role played by (optical) networks in supporting today's applications [7]. Cloud computing relies quite heavily on the concept of virtualization (cf. virtual machines): physical infrastructure is partitioned logically into distinct entities, such that applications can be run in virtually isolated environments while maximizing the use of physical resources by sharing them. This idea has also been adopted in networking: multiple virtual network operators (VNOs) share the same underlying physical infrastructure (i.e., fibers and optical cross-connects, OXCs, ROADMs), while VNOs only have access to (and full control over) their own resources.

In this paper, we focus on resilient virtual topology mapping: how to decide on what routes to follow in the physical network to map the virtual connections from source nodes to data centers where the applications are being served? The cloud services' requests are offered by a virtual network operator (VNO), who runs her virtual network (VNet) on top of the physical network resources offered by a physical infrastructure provider (PIP). The problem we address is how to determine a resilient VNet topology that minimizes the bandwidth resources that are requested by the VNO to the PIP, assuming time-varying traffic. We assume a VNO-resilience scheme, i.e., rerouting in the virtual network under the VNO control (see below, II-B, or, e.g., [8]). We design the VNet such that requests can survive single failures, which can each affect either the physical network or data center infrastructure.

B. VNO-resilience

The VNO-resilience model we adopt is exactly the same as in our earlier work [6], [8] and illustrated in Fig. 1: it provides 1:1 protection routing in the VNet for network failures, where the working and protection paths of a service have to be physically link/node disjoint. The working path π^{W} routes the services from their source node v_S towards the primary DC d_1, the protection path π^{B} towards the backup DC d_2, while π^{W} and π^{B} are disjoint in their physical layer mapping. In addition, a synchronization path π^{S} is established in order to handle migration and failure routing requirements when a DC failure occurs: services then need to be rerouted from primary d_1 to backup d_2. Thus, the resulting VNet for the request from source v_s comprises three virtual paths, mapped to resp. the physical $\pi^{\mathrm{W}}, \pi^{\mathrm{B}}$ and π^{S} paths. Note that

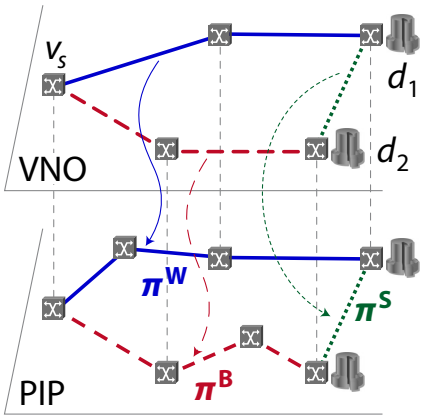

Fig. 1. The VNO-resilience scheme.

both π^{W} and π^{B} need to carry the overall traffic (but π^{B} only when π^{W} or d_1 are affected by a failure), but π^{S} possibly only a fraction thereof, since it is used just to keep the state at the backup location d_2 synchronized with that of d_1 (or vice versa) to allow smooth migration upon d_1 failure (or recovery).

Further, we assume that there is an automatic switch-back to the original network path and DC once a fault is repaired, and therefore we allow reusing the same network/DC capacity to protect against other failures: backup capacity is shared. Protection is guaranteed against any single link failure and any single DC failure, under the assumptions that (A1) the backup DC has a different location than the primary DC, (A2) π^{W} and π^{B} are link disjoint and, (A3) π^{W} and π^{S} are link disjoint.

We now qualitatively discuss the various failure cases we protect against:

(i) **Failure of link** $\ell \in \pi^{\mathrm{W}}$: the request is rerouted to the backup data center d_2, using the backup path π^{B} (which is link disjoint from π^{W}, thus $\ell \notin \pi^{\mathrm{B}}$). If $\ell \in \pi^{\mathrm{S}} \cap \pi^{\mathrm{W}}$, then as long as the failure is not restored, the primary data center d_1 cannot be kept in sync with the now operational d_2. Thus, right after the repair of ℓ, the primary d_1 is in stale state, and hence switching back to d_1 either suffers from this stale state or needs to wait some extra time to handle the requests again. The remedy is of course to enforce $\pi^{\mathrm{W}} \cap \pi^{\mathrm{S}} = \emptyset$. (Yet, note that the same issue of a non-synchronized primary d_1 clearly also occurs after the repair of d_1 that failed itself.)

(ii) **Failure of link** $\ell \in \pi^{\mathrm{S}} \setminus \pi^{\mathrm{W}}$: there is no immediate issue. Yet, if shortly after ℓ's repair, working path π^{W} fails, the switchover to the backup d_2 (via path π^{B}) suffers from stale state since the failing π^{S} interrupted the synchronization between primary and backup DCs. This can only be remedied by

providing a second synchronization path $\pi^{S'}$ that is link disjoint with π^S.

(iii) **Failure of link** $\ell \in \pi^B$: again no immediate problem arises (since this means that π^W is operational, given $\pi^W \cap \pi^B = \emptyset$). However, if $\ell \in \pi^S \cap \pi^B$ and shortly after ℓ's repair the primary path π^W (or d_1) fails — meaning that now π^B is followed towards d_2 — the secondary data center d_2 might not be fully synchronized yet. Clearly, this can be remedied by choosing $\pi^B \cap \pi^S = \emptyset$. Yet, the issue is similar to the one of case (ii), which obviously remains, even if we take $\pi^S \cap \pi^B = \emptyset$.

(iv) **Failure of primary DC** d_1: requests are rerouted to backup d_2 via the π^B path. Clearly, the failing d_1 cannot be kept in sync with the now operational backup d_2. Thus, we might need to wait some time after d_1's repair to switch back requests via π^W. Any failure that would occur shortly after d_1's repair and which would prevent services to remain being served at d_2 clearly could imply service degradation because of the unsynchronized d_1: (a) failure of π^S, (b) failure of π^B, or (c) failure of d_2. However, protection against such a failure event requires extra DC resources or extra paths.

C. Reconfiguration Scenarios for Time-varying Traffic

As in [6], we investigate whether it is worth reconfiguring the primary and the backup paths in order to save bandwidth when the communication traffic pattern changes. Note that this change is not necessarily limited to a scaling of the volume, but also its geographical pattern/distribution: large backbone networks (such as the ones that we are designing VNets over) might comprise different time zones where activities are shifted in time, and hence the resulting volume of cloud requests fluctuates differently.

Since changing the VNet mapping clearly may have an impact on the real-time performance of the cloud requests they are servicing, we propose to investigate three scenarios:

- In *Scenario I* (very conservative), we do not allow reconfiguring already established paths;
- In *Scenario II* we only allow reconfiguring backup and/or synchronization routes (π^B and/or π^S) for traffic that continues from one period to the next;
- In *Scenario III* we assume complete freedom and thus also allow to change the primary paths (π^W).

Whereas in [6] we developed a model to decide the transition from a single period to the next, we now present a model to jointly and globally decide on the routing in all time periods together.

III. Optimization Model

A. Notations

The cloud network is modeled by an undirected graph $G = (V, L)$ where V is the node set (indexed by v) and L is the link set (indexed by ℓ), for which $\omega(v)$ denotes the set of links incident with v.

We consider multi-period traffic, such that for each time period $t \in T$, the traffic is defined by the number of service requests (demands), originating from a set of source/service nodes $V_S \subseteq V$, with generic index v_S. We assume that requests originating from the same source node are aggregated, so that each request originating from source node v can be indexed by v and characterized by its bandwidth requirement $\Delta_{v,t}$ at time period t and δ_v (with $0 \leq \delta_v \leq 1$), representing the fraction of its demand that is required for synchronization between the primary and the backup data center. Note that to serve a request, it may be distributed over several DCs. We denote by $T^\star$ the set of all time periods excluding the first one.

Let $V_D \subseteq V$ be the set of data centers.

B. Configurations

The mathematical model we propose relies on the notion of configurations, where a configuration is associated with a set of service requests originating at a given source node. Let C be the overall set of configurations: $C = \bigcup_{v \in V_S} C_v$, where C_v is the set of configurations associated with source node $v \in V_S$. We define a configuration $c \in C_v$ by a set of 3 paths: (i) one primary path π^W originating at v towards a primary data center d^W, (ii) one backup path π^B originating at v_s towards a backup data center d^B, and (iii) one synchronization path π^S between the primary and the backup data center. We protect against single link failures as well as single data center failures.

More formally, in our mathematical model, a configuration is characterized by the given parameters:

- $p_\ell^{W,c}$ (resp. $p_\ell^{B,c}$, $p_\ell^{S,c}$) = 1 if link ℓ is used by the working (resp. backup, synchronization) path of configuration c, 0 otherwise;
- $a_v^{W,c}$ (resp. $a_v^{B,c}$) = 1 if node $v \in V_D$ is selected as the primary (resp. backup) data center, 0 otherwise.

We consider a decomposition model, following the column generation strategy. The so-called restricted master problem (RMP) assumes a given set of candidate configurations and chooses which ones to use as to satisfy the requested demands (in every period) with minimal bandwidth requirements. Given a (initial) solution, the so-called pricing problem (PP) generates a new configuration (if possible) that would allow to reduce the overall cost in the linear relaxation of the RMP if added to the set of possible configurations. There will be as many PPs as the number of source nodes. For $v \in V_S$, PP(v) finds a new set of working,

backup and synchronization paths for source node v and some chosen working and backup data centers. The new configuration is then added to the configuration subset that defines the RMP. Note that to derive the PP, we solve the Linear Programming (LP) relaxation of the RMP. In the end, when none of the PP(v) finds an new (improving) configuration, we solve the final RMP as an ILP.

Note that PPs for different source nodes can be solved simultaneously in parallel (based on the same RMP solution). By solving multiple PPs in parallel, the number of RMP iterations (with growing number of configurations) can potentially be reduced. We however leave the study of such parallelization strategies (e.g., how many PPs to solve in parallel, and how to cycle through the source nodes) out of scope for this paper. Suffice to note that for the experiments presented in Section IV, we solved all PPs (i.e., one PP for each source node) in parallel, thus potentially adding one extra configuration for each source node before solving the LP relaxation of RMP with the updated configuration set.

The master problem is described in detail next. The pricing problem (PP) follows quite straightforwardly from the RMP: the objective follows automatically (see the general column generation method, e.g., [9]), and the constraints amount to the classical flow constraints for each of the working, backup and synchronization paths, extended with the appropriate disjointness constraints (as discussed qualitatively in Section II-B).

C. Objective

We first define the variables:

- z_t^c: number of bandwidth units of the demand originating from v that is supported by configuration $c \in C_v$ at time period t.
- $\beta^{\mathrm{W}}_{\ell,t}$ (resp. $\beta^{\mathrm{B}}_{\ell,t}, \beta^{\mathrm{S}}_{\ell,t}$): required amount of bandwidth on link ℓ at time period $t \in T$, for provisioning the working (resp. synchronization, backup) paths.

The objective is to minimize the overall (working + backup + synchronization) bandwidth requirements:[1]

$$\min \quad \max_{t \in T} \quad \sum_{\ell \in L} \underbrace{\left(\beta^{\mathrm{W}}_{\ell,t} + \beta^{\mathrm{B}}_{\ell,t} + \beta^{\mathrm{S}}_{\ell,t}\right) \cdot \|\ell\|}_{\mathrm{BW}_\ell} \quad (1)$$

where $\|\ell\|$ represents the length of link ℓ and BW_ℓ is the total bandwidth cost for a given link ℓ.

D. Constraints

There are two sets of constraints. The first one aims at checking the demand requirements, and at computing the bandwidth requirements:

$$\sum_{c \in C_v} z_t^c \geq \Delta_{v,t} \qquad v \in V_{\mathrm{S}}, t \in T \qquad (2)$$

$$\sum_{c \in C} p_\ell^{\mathrm{W},c} z_t^c = \beta^{\mathrm{W}}_{\ell,t} \qquad \ell \in L, t \in T \qquad (3)$$

$$\sum_{v \in V_{\mathrm{S}}} \sum_{c \in C_v} \delta_v \, p_\ell^{\mathrm{S},c} z_t^c = \beta^{\mathrm{S}}_{\ell,t} \qquad \ell \in L, t \in T \qquad (4)$$

$$\sum_{c \in C} p_{\ell'}^{\mathrm{W},c} \, p_\ell^{\mathrm{B},c} \, z_t^c \leq \beta^{\mathrm{B}}_{\ell,t} \qquad \ell' \in L, \; \ell \in L \setminus \{\ell'\}, t \in T \qquad (5)$$

$$\sum_{c \in C} a_v^{\mathrm{W},c} \, p_\ell^{\mathrm{B},c} \, z_t^c \leq \beta^{\mathrm{B}}_{\ell,t} \qquad v \in V_{\mathrm{D}}, \; \ell \in L, t \in T. \qquad (6)$$

$$z_t^c \in \mathbb{R} \qquad c \in C, t \in T \qquad (7)$$

$$\beta^{\mathrm{W}}_{\ell,t}, \; \beta^{\mathrm{B}}_{\ell,t}, \; \beta^{\mathrm{S}}_{\ell,t} \in \mathbb{R} \qquad \ell \in L, t \in T. \qquad (8)$$

Constraint (2) guarantees that all bandwidth requirements will be satisfied in all time periods: the provisioned bandwidth of configurations rooted at node v satisfies the overall requested bandwidth $\Delta_{v,t}$ for traffic originating at v. Constraints (3) and (4) compute the working and synchronization bandwidth requirements on link ℓ during time period t, respectively. Constraint (5) (resp. (6)) ensures that the provisioned backup bandwidth on link ℓ during time period t suffices to carry the rerouted traffic under failure of a single link ℓ' (resp. of a single data center v). The last two set of constraints (7) and (8) define the domain of the variables.

The second set of constraints enforces routing constraints across time periods, in particular for Scenarios I and II. The constraints are, for all $v \in V_{\mathrm{S}}$, $\ell \in L$, $t \in T^\star$:

$$\beta^{\mathrm{W}}_{v,\ell,t} - \beta^{\mathrm{W}}_{v,\ell,t-1} \begin{cases} \geq 0 & \text{if } \Delta_{v,t} \geq \Delta_{v,t-1} \\ \leq 0 & \text{else} \end{cases} \qquad (9)$$

$$\beta^{\mathrm{B}}_{v,\ell,t} - \beta^{\mathrm{B}}_{v,\ell,t-1} \begin{cases} \geq 0 & \text{if } \Delta_{v,t} \geq \Delta_{v,t-1} \\ \leq 0 & \text{else} \end{cases} \qquad (10)$$

$$\beta^{\mathrm{S}}_{v,\ell,t} - \beta^{\mathrm{S}}_{v,\ell,t-1} \begin{cases} \geq 0 & \text{if } \Delta_{v,t} \geq \Delta_{v,t-1} \\ \leq 0 & \text{else} \end{cases} \qquad (11)$$

where

$$\beta^{\mathrm{W}}_{v,\ell,t} \triangleq \sum_{c \in C_v} p_\ell^{\mathrm{W},c} \, z_t^c,$$

and similarly for $\beta^{\mathrm{B}}_{v,\ell,t}$ and $\beta^{\mathrm{S}}_{v,\ell,t}$.

When the traffic volume originating at node v increases from one period to the next (i.e., $\Delta_{v,t} \geq \Delta_{v,t-1}$), constraint (9) ensures that on link ℓ the traffic portion from that node (i.e., $\beta^{\mathrm{W}}_{v,\ell,t}$) is at least the same as the period before: we are sure that all previous traffic can keep following the same routes. Conversely, if traffic decreases (i.e., $\Delta_{v,t} < \Delta_{v,t-1}$), we are sure

[1] Ideally, in case of ties, we should encourage to choose configurations that minimize the amount of routing changes from one period to the next: as a first priority, do not to change the working paths, and as a second priority do not change the backup/synchronization paths, in case the same routing paths can be (re)used in consecutive periods for requests that originate from the same source node. Explicit addition of these objectives is left for future work.

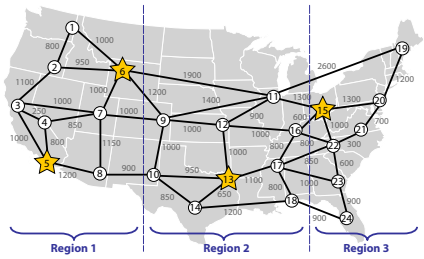

Fig. 2. USA network with our 3 regions. Yellow stars indicate the data centers, labels of links l their length $\|l\|$ in km.

that a previously unused link (i.e., $\beta^{\mathrm{w}}_{v,\ell,t-1} = 0$) will not be allowed to carry traffic in the subsequent period. Constraints (10) and (11) follow the same reasoning for the backup resp. synchronization paths.

Thus, for Scenario I, we enforce all three constraints (9)–(11). For Scenario II we only need to fix the working paths, hence only enforce (9). Finally, for Scenario III the traffic can be rerouted freely and thus we do not add any constraints beyond (2)–(8)).

IV. Numerical Results

A. Case Study Setup: Traffic and Topology

For our case study, we consider 3 different traffic volumes (A, B, and C) per time slot, that cyclically repeat: of all traffic requests that start in any of the three time slot types, 13% start in time period A, 38% in the second time period B, 49% in third time period C. We further assume three time zones (Regions), each shifted one time slot compared to the neighboring one: while the most western Region 1 goes through (A, B, C), simultaneously Region 2 goes through (B, C, A) while the most eastern Region 3 goes through (C, A, B). We distribute the total traffic volume, i.e., the total set of requests (over the whole day), over the three Regions proportionally to the number of network nodes in each region. As topology, we use the USA network illustrated in Fig. 2, where 33.33% of traffic requests originate in Region 1, 37.50% in Region 2 and 29.17% in Region 3. We will consider two cases:

- *Pattern #1*: 20% of requests in each time zone and each time slot just last two slots, while the other 80% last just for the single time slot where they start.
- *Pattern #2*: 80% of requests in each time zone and each time slot last two slots, 20% last just one.

B. Results: Bandwidth Savings by Rerouting

The relative change in bandwidth cost (i.e., the first summation of the optimization objective (1)) for the various scenarios is shown in Fig. 3. From these numerical results, we learn that the total bandwidth cost is reduced with on average 5.1% (resp. 6.4%) for Scenario II (resp. Scenario III) with traffic Pattern #1, and by 6.9% (resp. 8.2%) with Pattern #2 (where the average is taken over all traffic instances). This net saving mainly stems from a reduction of bandwidth for the backup paths, due to increased sharing: we noted an average reduction of the backup bandwidth cost with on average 11.5% (resp. 13.4%) for Pattern #1 and 14.2% (resp. 16.3%) for Pattern #2, for Scenario II (resp. Scenario III). We verified that these savings do not require all 2-period traffic requests to change their routing when going from one period to the next, but only about half of them. Further, these preliminary results suggest that the cost advantage can be achieved by only changing the backup/synchronization paths (Scenario II): there is only a limited advantage of allowing also the working path to be changed (Scenario III).

V. Conclusion

We studied the interest of re-provisioning the working and the backup paths in the context of resilient anycast routing traffic in cloud computing, assuming time-varying traffic, where the path provisioning can be updated periodically, assuming a time-slotted routing approach. We proposed a global optimization model, considering optimization of the routing over a set of multiple subsequent time slots in a single step. We propose a column generation model, using a path formulation in the master problem and pricing problems to find new (combinations of) paths. In an initial (small) case study, we note that the bandwidth cost savings may be reduced with up to almost 8% of the total cost, without needing to change all requests that survive multiple time slots (i.e., only about half of them need routing changes to realize these cost savings). Also, our small example suggests that the added value of also allowing the working routes (as opposed to only backup/synchronization routes) seems limited. Further experiments should confirm these hypotheses. Also, we suggest to evaluate the impact of the server locations (e.g., scattered vs. paired as in [10]), and investigate a broader range of time-varying traffic patterns.

Acknowledgment

B. Jaumard has been supported by a Concordia University Research Chair (Tier I) and by an NSERC (Natural Sciences and Engineering Research Council of Canada) grant.

References

[1] M. Gharbaoui, B. Martini, and P. Castoldi, "Anycast-based optimizations for inter-data-center interconnections," *Journal of Optical Communications and Networking*, vol. 4, no. 11, pp. B168–B178, 2012.

[2] C. Develder, J. Buysse, B. Dhoedt, and B. Jaumard, "Joint dimensioning of server and network infrastructure for resilient optical grids/clouds," *IEEE/ACM Transactions on Networking*, pp. 1–16, Oct. 2013.

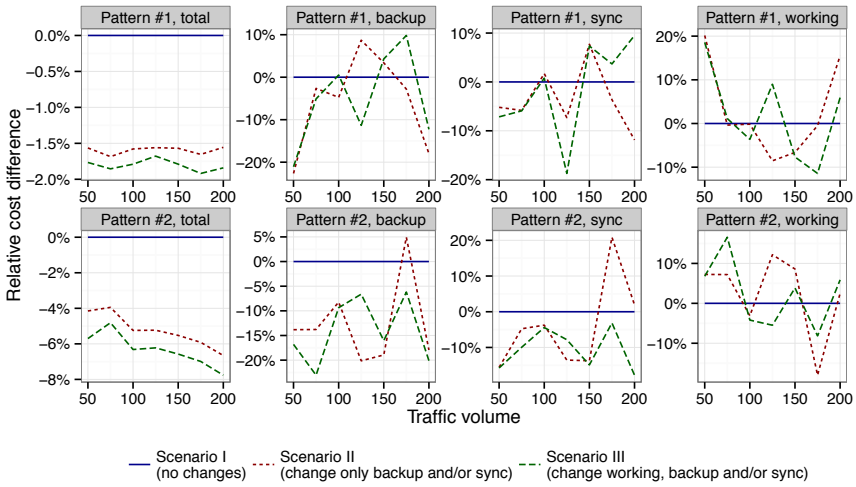

Fig. 3. Relative cost difference compared to not changing any routing from one period to the next (i.e., relative cost change compared to the corresponding Scenario I solution).

[3] X. He and G.-S. Poo, "Sub-reconfiguration of backup paths based on shared path protection for WDM networks with dynamic traffic pattern," in *The Ninth International Conference on Communications Systems (ICCS)*, 2004, pp. 391 – 395.

[4] S. Srivastava, S. R. Thirumalasetty, and D. Medhi, "Network traffic engineering with varied levels of protection in the next generation internet," in *Performance Evaluations and Planning Methods for the Next Generation Internet*, A. Girard, B. Sanso, and F. Vazquez-Abad, Eds. Springer Verlag, 2005.

[5] S. Sebbah and B. Jaumard, "Differentiated quality-of-protection in survivable wdm mesh networks using p-structures," *Computer Communications*, vol. 36, pp. 621–629, March 2013.

[6] M. Bui, T. Wang, B. Jaumard, D. Medhi, and C. Develder, "Time-varying resilient virtual network mapping for multi-location cloud data centers (invited)," in *Int. Conf. Transparent Optical Netw. (ICTON)*, Graz, Austria, June 2014, pp. 1–7.

[7] C. Develder, M. Leenheer, B. Dhoedt, M. Pickavet, D. Colle, F. Turck, and P. Demeester, "Optical networks for grid and cloud computing applications," in *Proceedings of the IEEE*, vol. 100, May 2012, pp. 1149 – 1167.

[8] M. Bui, B. Jaumard, and C. Develder, "Anycast end-to-end resilience for cloud services over virtual optical networks (invited)," in *Int. Conf. Transparent Optical Netw. (ICTON)*, Cartagena, Spain, June 2013, pp. 1–4.

[9] V. Chvatal, *Linear Programming*. Freeman, 1983.

[10] M. Bui, B. Jaumard, and C. Develder, "Resilience options for provisioning anycast cloud services with virtual optical networks," in *IEEE International Conference on Communications - ICC*, Sydney, Australia, June 2014, pp. 3462 – 3468.